Traditional Set Operations

Detailed Syllabus

- Union
- Intersection
- Difference
- Cartesian Product

Relational Algebra

- **Relational algebra is a formal language associated with the relational model.**

- **Informally, relational algebra is a (high-level) procedural language.**

Relational Algebra

- **Relational algebra operations work on one or more relations to define another relation without changing the original relations.**

- **Both operands and results are relations, so output from one operation can become input to another operation.**

- **Allows expressions to be nested, just as in arithmetic. This property is called closure.**

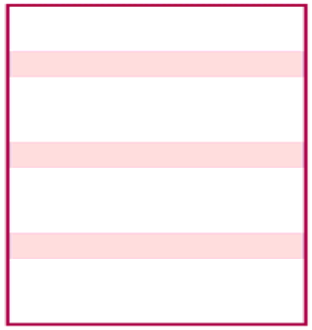
Relational Algebra

- **Five basic operations in relational algebra: Selection, Projection, Cartesian product, Union, and Set Difference.**

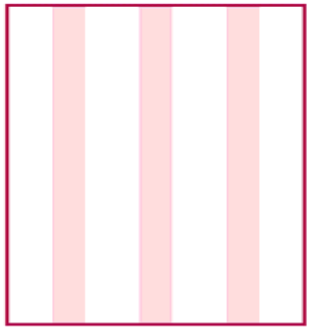
- **These perform most of the data retrieval operations needed.**

- **Also have Join, Intersection, and Division operations, which can be expressed in terms of 5 basic operations.**

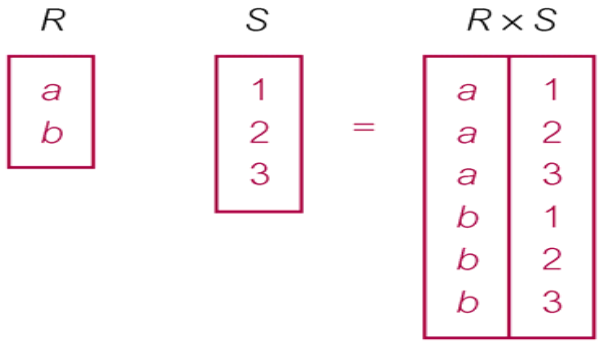
Relational Algebra Operations



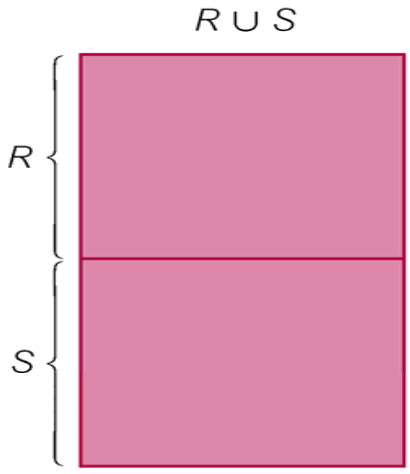
(a) Selection



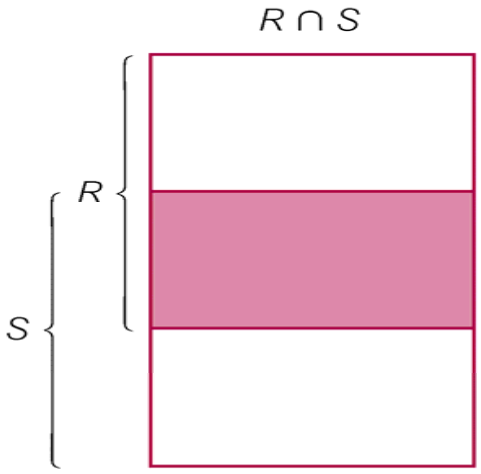
(b) Projection



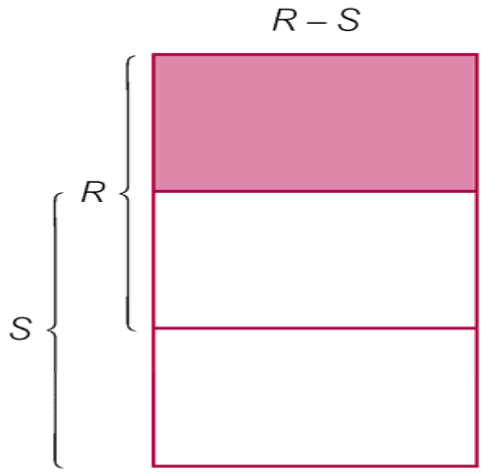
(c) Cartesian product



(d) Union



(e) Intersection



(f) Set difference

Sequence of Operations

- We may want to apply several relational algebra operations one after the other.

  e.g. Retrieve the first name, last name and salary of all employees who work in department no. 5

  We can write the operation as a single relational algebra operation by nesting the operations

  e.g.

  πFname,Lname,Salary (σDno=5 (Employee))

Sequence of Operations

- Or we can apply one operation at a time and create intermediate result relations.

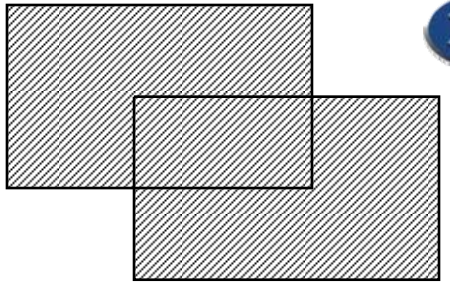
DEP5_EMPS ← (σDno=5 (Employee))

RESULT ← πFname,Lname,Salary(DEP5_EMPS)

Union

- $R \cup S$
  - Union of two relations R and S defines a relation that contains all the tuples of R, or S, or both R and S, duplicate tuples being eliminated.
  - R and S must be union-compatible.

- If R and S have *I* and *J* tuples, respectively, union is obtained by concatenating them into one relation with a maximum of $(I + J)$ tuples.



Union

Student

Fname	Lname
Kapila	Dias
Nimal	Perera
Ajith	Silva
Rohan	Mendis

Instructor

FN	LN
Sunil	De Silva
Kamal	Soysa
Saman	Silva
Kapila	Dias
Nimal	Perera

Stu-Inst

Fname	Lname
Kapila	Dias
Nimal	Perera
Ajith	Silva
Rohan	Mendis
Sunil	De Silva
Kamal	Soysa
Saman	Silva

$$\text{Stu-Inst} = \text{Student} \cup \text{Instructor}$$

Example

- Retrieve the EmpNo of all employees who either work in department 5 or directly supervise an employee who works in department 5.

Example

DEP5_EMPS ← (σDno=5 (Employee))

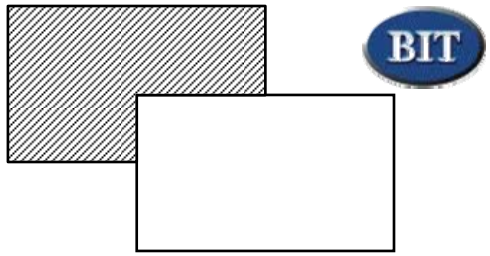
RESULT1 ← πEmpNo(DEP5_EMPS)

RESULT2(EmpNo) ← πSuperNo(DEP5_EMPS)

RESULT ← RESULT1 ∪ RESULT2

Set Difference

- R – S
  - Defines a relation consisting of the tuples that are in relation R, but not in S.
  - R and S must be union-compatible.



Difference

Student

Fname	Lname
Kapila	Dias
Nimal	Perera
Ajith	Silva
Rohan	Mendis

Stu-Inst

Fname	Lname
Ajith	Silva
Rohan	Mendis

Stu-Inst = Student - Instructor

Inst-Stu = Instructor - Student

Instructor

FN	LN
Sunil	De Silva
Kamal	Soysa
Saman	Silva
Kapila	Dias
Nimal	Perera

Inst-Stu

Fname	Lname
Sunil	De Silva
Kamal	Soysa
Saman	Silva

Intersection

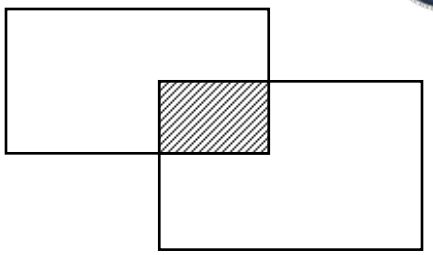
- **R ∩ S**
  - **Defines a relation consisting of the set of all tuples that are in both R and S.**
  - **R and S must be union-compatible.**

- **Expressed using basic operations:**

  R ∩ S = R – (R – S)

Student

Fname	Lname
Kapila	Dias
Nimal	Perera
Ajith	Silva
Rohan	Mendis



Intersection

Instructor

FN	LN
Sunil	De Silva
Kamal	Soysa
Saman	Silva
Kapila	Dias
Nimal	Perera

Stu-Inst

Fname	Lname
Kapila	Dias
Nimal	Perera

Stu-Inst = Student ∩ Instructor

Union, Intersection and Difference

- Both Union and Intersection are commutative

  $R \cup S = S \cup R$ and $R \cap S = S \cap R$

Either operation can be applied to any number of relations

$R \cup (S \cup T) = (R \cup S) \cup T$ and

$(R \cap S) \cap T = R \cap (S \cap T)$

Difference operation is not *commutative*

$R - S \neq S - R$

Cartesian product

- **R X S**
  - Defines a relation that is the concatenation of every tuple of relation R with every tuple of relation S.

  - Creates a single table from two tables.

  - If R has n_R tuples and S has n_S tuples then R X S will have $n_R * n_S$ tuples.

Employee

E-No	E-Name	D-No
179	Silva	7
857	Perera	4
342	Dias	7

Department

D-No	D-Name	M-No
4	Finance	857
7	Sales	179

Emp-Info

E-No	E-Name	D-No	D-No	D-Name	M-No
179	Silva	7	4	Finance	857
857	Perera	4	4	Finance	857
342	Dias	7	4	Finance	857
179	Silva	7	7	Sales	179
857	Perera	4	7	Sales	179
342	Dias	7	7	Sales	179

Emp-Info = Employee × Department